I-FRAME de VE3PKT

.--. -.- -

Newsletter of the

HAMILTON & AREA PACKET NETWORK

Bulletin number 3.
April, 1981.

Great news! The Hamilton and Area Packet Network station node now has a licensee! VE3MWM (alias Stu Beal) passed his amateur code exam at 4 minutes to midnight, so to speak. Hopefully, he will now have some time to devote to the implementation of the station controller software for the custom card.

This issue will be somewhat skimpy by the standard of our previous publications but contains some good stuff: a continuation of the packet-related bibliography, John(DVV)'s trap dump formatter, a minimum standard for information exchange from end to end (a TIP standard), and a couple of packet related papers from the literature.

Another possible Network in Burlington

Initial contacts have been made here in Burlington, with the Burlington MicroComputer Club, to set up a second network of nodes using the General Radio Service frequencies. Since the club offers an additional pool of talent, we thought we might get some of the members interested in the communication field. Bill Bouwhuis of the local DOC office indicated that AFSK would be alright in CB as long as the standard rules were followed! 4 watts max. carrier power, voice identification, listen before transmission and no intentional interference with other users. In addition, if the Department received any formal complaints, the mode might be terminated.

The last point obviously precludes the users investing in expensive hardware initially, since if shut down, the units would be of little use. A decision to use a start-stop protocol to begin with was therefore made, with the opportunity to move up to VADCG boards after the net had been on the air several months.

The net will have reasonably localized coverage but will offer the chance to make realistic measurements of throughput and vulnerability in the presence of jamming and other antagonistic forces.

The GRS net could be provided with a gateway to the VHF net at a later date if the users on both sides were in favor.

HAPN Meeting

The second meeting of the HAPN occurred on March 8th, again at CCIW, with a smaller turnout. Discussion items were reasonably wide ranging but a major item decided on was the minimum capability of the TIP. A report of the meeting will be found later in this issue.

AMRAD

This group was recommended by Doug Lockhart of VADCG and when contacted, sent out a bunch of their February newsletters to be distributed to the meeting March 8th. Unfortunately, they didn't arrive until after the event, so I am including a copy in this mailing, along with an application form.

The group is quite interested in things digital and has Special Temporary Authority from the FCC to experiment with packet and spread-spectrum modes.

BUGS

The effect of a bug relating to full LIP buffers has been characterized and it is expected there will be a patch shortly. The error manifests itself if the TIP doesn't drain the LIP buffers fast enough and a Receive Not Ready (RNR) is sent. It seems the Poll bit is set in the RNR and when the response comes from the secondary, it is rejected with a RNR with poll bit on, and so on ad infinitum.

Proposed Minimum TIP Capability

A group of members came up with a suggestion at the last meeting regarding what the TIP should handle as a minimum, so all users can be brought to a common level initially.

It is expected that this minimum will be upgraded in the future as we get more and more familiar with the requirements of the applications we will be running.

The minimum character set which must be handled by a TIP is that represented by the ASCII characters with Hex codes between 20(16) and 5A(16) plus the characters 0A(16) [LF] and 0D(16) [CR]. All other characters may be treated in any (as yet unspecified) way by the TIP at the far end, ranging from true interpretation to discard.

In summary:

Hex	Graphic	Hex	Graphic	Hex	Graphic
0A	LF	32	2	46	F
0D	CR	33	3	47	G
20	SPACE	34	4	48	H
21	!	35	5	49	I
22	"	36	6	4A	J
23		37	7	4B	K
24	$	38	8	4C	L
25	%	39	9	4D	M
26	&	3A	:	4E	N
27	'	3B	;	4F	O
28	(	3C	<	50	P
29	)	3D	=	51	Q
2A	*	3E	>	52	R
2B	+	3F	?	53	S
2C	,	40	@	54	T
2D	-	41	A	55	U
2E	.	42	B	56	V
2F	/	43	C	57	W
30	0	44	D	58	X
31	1	45	E	59	Y
				5A	Z

Additions to Mailing List

The following name has been added to the mailing list:

Mr.William M. Moran, W4MIB,
109 Mill St., N.E.,
Vienna, Virginia,
22180.

Bill heard about the organization thru the AMRAD newsletter of April, 81. In fact, I received his check even before I got the newsletter!

Bibliography, continued.

This bibliography was made by searching the last 6 months of Current Contents, Engineering, Technology and Applied Sciences for the words ALOHA, Network, Packet or Protocol. The number after the sequence number is the issue number of CC the citation was found in.

I have begun to include papers on communication techniques as well since some of our members expressed an interest.

1. 81-7
A network oriented information structure: networks, semantics and structures.
Computers and Graphics, v 5 n 2-4 p 41

2. 81-7
Packet system puts local networks on cable tv
Electronic Design v 28 n 21 p 42

3. 81-8
Network problem-determination aids in microprocessor-based modems
IBM Journal of research and devel. v 25 n 1 p 3

4. 81-8
a 2400 bit/s microprocessor based modem
IBM Journal of research and devel. v 25 n 1 p 17

5. 81-8
improved decoding scheme for frequency hopped multi-levelfsk system
Bell system technical journal v 59 n 10 p1839

6. 81-10
Congestion control of packet-switched networks with 3 types of input buffer limits.
Computer Communications v 3 n 6 p 270

7. 81-10
Configuration dependent performance of a prioritized CSMA broadcast network
Computer v 14 n 2 p 51

8. 81-10
an efficient design of large scale communication networks with a decomposition technique
IEEE trans Circuits and Systems v 27 n 12 p 1169

9. 81-11
Analytical study of multi-class queueing networks with variable routing
RAIRO Recherche operationnelle/operations research v 14 n 4 p 331

10. 81-11
A queueing system in which customers require a random number of servers
Operations Research v 28 n 6 p 1335

11. 81-11
An overflow system in which queueing takes precedence
Bell syst. tech J. v 60 n 1 p 1

12. 81-11
Fundamental limits in information theory.
Proc IEEE v 69 n 2 p 239

13. 81-11
Audio and Video bit-rate reduction
Proc IEEE v 69 n 2 p 252

14. 81-11
Realization of a decoder for two MFM codes
Electronics Letters, v 17 n 3 p 117

15. 81-11
Information theory and Communication Theory
Frequenz v 35 n 1 p 2

16. 81-11
Bit errors in the Digital communications network - a general survey
Frequenz v 35 n 1 p 9

17. 81-12
Distributed algorithms for generating loop-free paths in networks with frequently changing topology
IEEE Trans Communic. v 29 n 1 p 11

18. 81-12
Feedback communications over fading channels
IEEE Trans Communic. v 29 n 1 p 50

19. 81-12
Block error probability for noncoherent FSK with diversity for very slow Rayleigh fading in gaussian noise
IEEE Trans Communic. v 29 n 1 p 57

20. 81-12
An improved stop-and-wait ARQ logic for data transmission in mobile systems
IEEE Trans Communic. v 29 n 1 p 68

21. 81-12
An optimal algorithm for mutual exclusion in computer networks
Comm ACM v 24 n 1 p 9

22. 81-12
EDITORIAL: Legislating for Frequency Conservation
The Radio and Electronic Engineer v 51 n 2 p 51

23. 81-13
Distributed communication architecture forms framework for network design
Computer design v 20 n 2 p 121

24. 81-13
A unified approximate evaluation of congestion functions for smooth and peaky traffics
IEEE trans Communic. v 92 n 2 p85

25. 81-13
Effects of packet losses in waveform coded speech and improvements due to an odd-even sample interpolation proceedure
IEEE trans Communic. v 92 n 2 p101

26. 81-13
Computer simulation results on frequency hopped MFSK Mobile radio- noiseless case
IEEE trans Communic. v 92 n 2 p125

27. 81-13
Buffer sizing of a packet voice receiver
IEEE trans Communic. v 92 n 2 p152

28. 81-13
Performance of forward error correction and feedback coding schemes
IEE Proc - F v 128 n 1 p 15

29. 81-13
Channel estimation for an HF radio link
IEE Proc - F v 128 n 1 p 33

30. 81-14
MSK and QPSK modulation with bandlimiting filters
IEEE Trans Aerospace and electronic syst. v 17 n 1 p 2

31. 81-14
Digital radio systems; Part I - status; application and performance of medium capacity systems
Austral. Telecomm. Research v 14 n 2 p 51

32. 81-14
A proof of Marton's coding theorem for the discrete memoryless broadcast channel
IEEE trans Inform. Theory v 27 n 1 p 120

33. 81-14
The capacity region for the deterministic broadcast channel with a common message
IEEE trans Inform. Theory v 27 n 1 p 122

34. 81-14
A new method for optimum alignment of digital communication equipment
IEEE Trans. Instrum. Measurm. v 30 n 1 p 12

AMRAD NEWSLETTER

Amateur Radio Research and Development Corporation

February 1981

OUR FEBRUARY 2 MEETING will be a talk by Bob Watson on radioteletype demodulation techniques. Bob is an electronic engineer with Watkins-Johnson and works with state-of-the-art demodulation circuitry. The meeting will be held at the Patrick Henry Branch Library, 101 Maple Ave E, Vienna, VA at 7:30 p.m. Visitors and guests are invited.

RTTY modulation and demodulation are of special concern to us at this time as we consider how best to transmit packet radio as part of an amateur network. A number of hams have just assumed that changing from Baudot to ASCII means an automatic increase of speed from 45 to 110 baud, thus a net loss in readability. Well, there's a lot more to it than that, and Bob may try to convince us that there could be a net gain if we use the built-in structure of ASCII (e.g., parity) and use optimal demodulation techniques.

IF YOU MISS THE MARCH NEWSLETTER ISSUE or if it's delayed in getting to you, it may be because your editor is traveling to Italy on business. If an issue is missed, we'll extend everyone's subscription accordingly. The trip is planned, but the exact date is still up in the air. So that you will know what is planned for future AMRAD meetings, here are the plans:

The March 2 meeting will be a return engagement by Ken Coghill, WB4ZOH with even more spectacular computer music. It is always a full house at Ken's concerts so plan to get to the meeting early to assure yourself of a seat. Bring the music lovers in your family.

Our April 6 meeting will not be held at the Patrick Henry Branch Library in Vienna but will be the 1981 ARRL Technical Symposium run by AMRAD at the Washington Hilton Hotel. The 6th is the beginning of the IEEE Vehicular Technology Society Annual Convention. By the way, we're still looking for technical papers for the symposium. If you can give a paper please contact Paul Rinaldo W4RI, 703-356-8918, days or evenings.

A COPY OF THIS NEWSLETTER is being sent to a number of individual amateurs who responded to an item in Ted Cohen's column in *CQ Magazine* about the proposed amateur computer network AMNET. An outline of the AMNET concept and some of the AMRAD activities in that direction is given in this newsletter.

THE VIENNA WINTERFEST™ is scheduled for February 22. AMRAD is planning to have two tables reserved for the club members' use. So, if you have just an armful of junk or less to peddle, bring it along to the AMRAD table. Perhaps you can spend an hour or so at the table when your feet give out.

THE SPREAD SPECTRUM SPECIAL INTEREST GROUP is now standing by to hear some word from the Federal Communications Commission on the AMRAD request for a Special Temporary Authority to do some experimenting. To date 29 stations throughout the U.S. have signed up to be participants. At this point, we are not looking for others to throw their hats in the ring. There may be opportunity for that after the STA is underway. The latest word from the FCC is that the request is working its way through the system and has been placed on the Interagency Radio Advisory Committee (IRAC) because any new amateur use of the 420-MHz band requires IRAC coordination. Our SSSig secretary, Hal Feinstein, WB3KDU is trying to keep the flow of information going by corresponding with individuals in the SSSig. We're also planning to set up a voice net at 4 p.m. EST on Sundays on 20 meters and simultaneously (QSP) on the AMRAD 2-meter repeater. The tentative 20-meter frequency is 14290 kHz.

THE ARRL PETITION OF DIGITAL TECHNIQUES that appeared in the AMRAD Newsletter two months ago has been assigned the designation RM-3788.

COMPUTING AND THE HANDICAPPED is the subject of several articles in IEEE Computer Society's *Computer* magazine for January 1981. Single copies are available from IEEE Service Center, 445 Hoes Ln, Piscataway, NJ 08854; $3 members, $6 non-members.

CORRESPONDENCE:

4211 7th Avenue
Rock Island, IL 61201
1 Dec 80

Dear Paul,

I have been a member of AMRAD for several years now. I thoroughly enjoy the newsletter. Living out here in the midwest and away from the "action," I rely on newsletters like yours to keep me abreast of what is going on in computers, etc.

One comment: Where did you get that list of Ham nets published in the Nov issue? I am a Digital Group owner and have not heard of the Saturday afternoon net. I have listened for 3 weeks and heard nothing.

I envy the calibre of programs presented at your meetings. I have given considerable thought to how groups such as yours can share their knowledge with others. My latest idea is to circulate video recorder tapes. VTR's are becoming more and more popular and hopefully they are available in your club. I propose that your group tape the programs and then circulate them to whomever would like to see them. Professional quality would not be necessary, only place a camera in the corner and turn it on and let the speaker present... I don't think that your club would become rich by offering tapes but it would surely be a service to members who are out of town and unable to attend meetings. I have checked with several local VTR dealers and they feel the VHS format is the most popular throughout the country. I therefore recommend adoption of the VHS format (besides, I just bought a VHS machine) hi.

I don't consider this to be a one way street. I would hope that the idea would catch on and clubs throughout the country could exchange tapes. However, I feel that a club such as AMRAD should pioneer such a program.

73,
John Greve, W9RI

Some time ago, we produced one tape on the AMRAD CBBS. It showed on a number of ATV repeaters, at some conventions and club meetings. We were hoping to do some more but were tied up in so many other things that it got sidetracked for a while. Anyway, your letter helped to push us back on track. We have a 10-minute tape produced by Applied Physics Lab of Johns Hopkins university which explains the national competition for computer aids to the handicapped. We taped the talk at the January meeting which featured Larry Bates, KØLB on S-100 computers. My talk at the Metrovision Club meeting on December 10 was taped by W9GVK of Metrovision. The talk was on spread spectrum. We hope to work closely with Metrovision on taping more meetings and other shows for distribution around the country.

As far as I know the DG net info is valid. I checked with W4MIB who said that he heard the net since the info appeared in the newsletter. Ed.

EQUIPMENT FOR SALE:

1. One KSR-33 TTY good condition. I upgraded to a real live printer and must sell this to help pay for the new one. This Teletype is serial, 20-mA current loop, like most TTY's. It is also a table-top model, no stand. Good deal, only $250.

2. Intel multibus computer system components (like new condition)
   a. One 4-slot multibus motherboard
   b. One National, 16 KB dynamic memory board (multibus)
   c. One Intel - SBC 80/20 - 4 single board computer (Intel multibus of course) This has 4 KB of static RAM, 4 EPROM sockets for either 2708 or 2716 EPROMS (4 or 8 KB).
      I/O - 2 - 8255 PIO's
      1 - 8251 serial port
      1 - 8253 CTC chip, 2 port, one used for baud rate, the other available.
   d. Documentation for all of this.
   e. 2 sets of monitor ROMs, one set has an assembler.
   f. Schematic for a multibus floppy disk drive controller using the 8271 chip (single density).
   New value for all of this approx $1900. Good deal for this package $600. This makes an excellent starter system that is expandable to extension buses, multi-processors, including the 8086. I also have an 8271 FDC chip available (separate arrangement).

3. Processor Tech - 4KRA 4K RAM board for the S-100 bus. 450 ns 2102 RAM chips, well burned in, works reliably. Complete with documentation $50.

Steve T. Stolen, eves 703-360-3470 Alexandria; days 202-695-3871 (leave message if I'm not in).

WESTLINK IS FACING A SEVERE money crunch, and for the first time is actively soliciting financial support from those in the Amateur Radio community who use the tape-recorded broadcasts of Amateur news. Until now all Westlink costs had been covered by owner-producer WA6ITF and production coordinator KH6IAF, but the telephone and other expenses have increased to the point where they can't afford to cover them from personal funds.

A Trust Has Been Set Up to assist in Westlink funding, to be administered by Dr. Norm Chaplin, K6PGX. Contributions made out to "Westlink Radio Network" can be sent to Norm care of Box 463, Pasadena, California 91102. Thanks *HR Report*.

R.W. ELECTRONICS, 3165 N Clybourn, Chicago, IL 60618 is offering a bare board & documentation for a Z-80 single-board computer for $49.95; assembled & tested $199.95.

PROTOCOL

David W. Borden, K8MMO
Rt 2, Box 233B
Sterling, VA 22170
703-450-5284 (Metro No.)

This month, Dave is again traveling so I have pulled tog ther some information of interest to amateur computer networking. Ed.

First, a letter from Hank Magnuski, KA6M:

311 Stanford Avenue
Menlo Park, CA 94025
December 31, 1980

Dear Dave,

It was a pleasure to talk to you recently about packet radio activity and plans, and as promised, here is a rundown on some of the work being done in the San Francisco Bay area:

First, I would like to announce the birth of what may be the first all digital, HDLC oriented, packet repeater in the U.S. This blessed event occured December 10th when KA6M/R went live on 2 meters running at 1200 baud. The repeater serves a dual function of both being a repeater and a beacon which can be used to test and check station equipment and programs. The heart of the repeater is a Z-80 microcomputer on an STD Bus card driving a custom built I/O board containing a Western Digital 1933 HDLC chip. The control program is written in Pascal/Z with assembly language interfaces for interrupt processing, and only requires 4K bytes of EPROM (two 2716's). The HDLC card interfaces to a 202 modem, which in turn drives an FT-202 transceiver. The FT-202 is a small crystal controlled transceiver which has solid-state transmit/receive switching.

The framing used for the packets is standard HDLC format with NRZI encoding. A packet consists of an address byte, a control byte and an information field with a maximum length of 256 bytes. The repeater checks the incoming packet for proper address range and CRC, and then repeats the packet with its own address byte and control field appended to the beginning of the packet. The original packet may be recovered at the receiver by stripping off the two added bytes.

The existence of the repeater, even though it is experimental, has started a snowball of packet activity in this area. Four of the Vancouver boards have been purchased and are very near completion. We have started a project to design an S-100 interface for the Western Digital chip and may possibly interface it to other computers as well.

Most of the hams in this area will be using 202 style modems for their initial attempts at ASCII and packet radio. In a moment of madness I purchased 54 Rixon/Sangamo T202C and T202D modems which were being dumped by an industrial broker, and then started seeding the Bay area with them. *(See article from the Homebrew Computer Club Newsletter reprinted in this newsletter. Ed.)* About half the stock is gone now, and purchasers include many members of the local RTTY repeater group, a 450 MHz repeater group, and various individuals including the editor of the Homebrew Computer Club newsletter, who is also a ham.

We now have a very nice RTTY mailbox system running here thanks to the work of Dave Alterkruse, W6RAW, and his helpers from the K6GWE/R Amateur Communications Society, the group which runs the 2-meter RTTY repeater. The program runs on a Northstar and is written in PL/1.

Future plans include a move to 450 or higher, and implementation of direct digital modulation as is being done in Ottawa. My approach parallels much of the work being done by the VADCG, and I hope to coordinate with them and eventually establish a link up the West Coast. I think Doug Lockhart and his associates there deserve credit for a lot of good ideas.

Finally, work on the AMSAT/AMICON spec will resume soon now that a launch date has been set. Any readers who desire a copy of the next draft should send me a note. Particular thanks go to Bob Carpenter for his comments on the first draft.

A number of issues have come up which deserve attention, and perhaps some members of AMRAD may have thoughts on these subjects:

Issue #1 - When is a repeater a repeater?

Is a simplex packet repeater or digipeater a "repeater" as defined in the FCC

rules? The answer to this question is crucial because it determines what frequencies may be used in packet radio service. For example, it would be nice to use the non-satellite portion of 145.5 - 146.0, but "repeaters" are not permitted there. It would also be nice to be able to use digipeaters on HF.

A conservative opinion goes something like this: Of the various types of stations and services defined in the rules, "repeater" clearly is the best match, and thus digipeaters must be in the repeater subbands.

A moderate opinion: The FCC never envisioned digipeaters and the intent of the rules really is to control duplex machines. Use any frequency.

A radical opinion: A digipeater is a remotely controlled object, a store-and-forward message box. The use of the word "repeater" in "packet repeater" is an unfortunate accident and should not be confused with the meaning of "repeater" as in the rules.

Issue #2 - What frequencies?

We should establish national frequencies for packet radio service. This issue has been submitted to the local frequency coordination council and may work its way up the bureaucracy. Has AMRAD chosen any simplex frequencies, and if so, what are they?

Issue #3 - Coordination

In order to form cross-country packet radio nets, we need a coordinating committee to identify and keep track of local and regional activities. Active and interested individuals need to be able to get together, missing sites in repeater links need to be identified, protocol problems have to get resolved, etc. For a starter, one person in each time zone should be identified and given a suitable title.

So there you have the status of Pack

Best regards to all for 1981,

Hank Magnuski, KA6M

Our thanks to Hank for the above informative letter. Now that the channel is open we hope to work as closely as possible with Hank toward establishing a North American amateur computer network.

The following items concern experimental 30-meter operation by VE3QB and VE3BPD:

30-METER BAND OPERATION was begun last week by VE3QB, using his recently assigned Canadian commercial callsign, VE9LFZ, for preliminary one-way transmissions with partner VE9LIN (VE3BPD). They plan to operate daily at 1500, 1700, and 1900Z plus random times evenings starting at 2200Z. CW and (later) ASCII will be the modes, using 10.101 and 10.149 MHz.

VE9LFZ Was Solid Copy on 10.101 MHz in the Midwest Thanksgiving afternoon, with a more than hour-long transmission that peaked as high as 5-6 even though Larry was running only five watts to an inverted V. Note: These are test transmissions only, authorized by the Canadian government, and Amateur operations on this band will be illegal untill January, 1982. *HR Report December 5, 1980.*

VE9LFZ and VE9LIN have been quite active on 30 meters, though they've had to reduce power to a watt or less to keep signals over the 300-mile path between them to levels appropriate for their experiments. Even with that low power, however, they've been good copy near Chicago on 10.149 MHz. *HR Report December 19, 1980.*

Here's a letter from VE3DPB:

November 24, 1980

Dear Paul,

Thanks for your letter. Yes, I missed the AMSAT annual meeting this year. I don't like missing them but sometimes it has to happen.

As you know I am very interested in methods of narrowing digital modulation receiving filters. In my mind the conventional CW filter in amateur receivers is way too wide for the baud rates we use but, of course, we are limited by the ringing problem.

Don (W3QVC) has done some interesting experiments, and we have given talks at the Dayton Hamvention (twice).

Larry VE3QB in Ottawa has become interested in Petit's system and suggested running some tests on the 10.1 MHz band. With this in mind we applied for permits and so far I have got mine with the call sign VE9LIN. Max. pwr. 8 watts and good for 1 year.

Anyway Paul, when we have something to report you can count on hearing from us. I think a lot of AMRAD and I only wish there were more such organizations.

73,
Bert, VE3DPB

And now a letter from VE3QB:

19801229

Dear Paul:

... this letter is to seek some information and to let you know what we are doing here in Ottawa.

First I have tried your bulletin board 703-281-2125 quite a number of times some months back. I always got an answer and data tones - but regardless of parity, speed, character sent I couldn't get it to

talk to me. Have you any further info on this? *(Yes. I don't know what the trouble was because lots of others have been successful accessing it. However, it is all moot now because we have moved our bulletin board to 703-734-1387. See the back page of the newsletter for info. Ed.)*

I am interested in your HEX program and would appreciate further details.

I've also been involved with our local bulletin board sponsored by the Ottawa Computer Group. I've written all the software and hope to have bubble mass storage by next summer. We have just finished a clock/calendar and I am now going to start a user selectable subsets of the stored information. The OCG have decided they will NOT support mechanical mass storage - i.e., floppies - so our two way messaging will have to wait until we get some bubbles, Hi.

For full computer conferencing we will link our bulletin board (613-236-3009 by the way - several CR's at 110 or 300 bps) to a large 370 type mainframe using our packet radio system. Our packet system is on 222.34 and runs at 9600 bps.

We are just getting ready to open up a gateway to HF from our local packet system. This has been much delayed but we will very shortly be on 14076 F1, 170 Hz. This is a fully automated system - no operator is even possible at this point. As well we are on 10.149 using QRP (commercial calls VE9LFZ (me) and Bert VE9LIN) developing computerized digital filters at low bit rates (10 bps & slower).

I have to run now but lets hear from you. I'll call in on your bulletin board if I get the info.

73,
Larry Kayser, VE3QB
24 Arundel Ave
Ottawa, Ontario
Canada K1K ØB6

AMNET

A NUMBER OF PEOPLE have written us in response to an item which appeared in Ted Cohen's column in *CQ* magazine. We have had some stimulating conversations with interested individuals over the past few months. As a result, the AMNET concept is continuing to evolve. We have developed a working philosophy along these lines. There *is* going to be an amateur computer network of some type springing to life sometime during the 1980's. The evidence is quite clear from the dynamic work undertaken by groups in Vancouver, Montreal, Ottawa, Chicago, San Francisco Bay area, and of course some meager efforts of our own. Also, there are roughly 200 computerized bulletin board systems (CBBS's) in the U.S. and Canada which represent a resource just crying to be tied together in a network. There have been years of discussions on how to pay for long distance calls between CBBS's - a sticking point. There has also been much debate about protocols. A number of the computer hobbyists interested in telephone linking of computers tend to dismiss amateur radio as a suitable way of moving personal computer traffic due to licensing requirements. The counter-argument is that there are enough licensed hams around North America to make a network work. Also, a sufficient number of the true prime movers in personal computing are already hams. So, it seems reasonable to use the Amateur Radio Service and Amateur Satellite Service frequencies where possible and supplement this with telephone calls. Cost of long-distance calls can be reduced by using higher speeds, calling after 11 p.m., and possibly using Value Added Networks or the so-called private phone companies that offer low long-distance rates between cities.

So, the basic idea is to have a number of nodes (computers) spread around North America, ham radio and/or telephone circuits to link them together, and an agreed protocol. Once you get much beyond that point, there may not be much agreement by the various advocates of amateur networking.

AMRAD has been in touch with many of the people in the U.S. and Canada who are trying to make serious contributions to amateur networking. Some concepts are taking shape but need much more discussion and coordination.

We have been talking to Telecommunications for the Deaf, Inc. (TDI) to see how some of the needs of the deaf could be met by a tie-in with an amateur network. (See Barry Strassler's column.) Hams are used to handling third party traffic. One advantage with the deaf is that they can originate and receive traffic in digital form, albeit in Baudot code. AMRAD's HEX system is a bulletin board with ability to speak either ASCII or Baudot. So, the code conversion problem can be solved by bilingual CBBS's such as HEX.

AMRAD has been working closely with the American Radio Relay League (ARRL) to get them to make a committment to amateur computer networking. The ARRL is a sizeable organization with resources that would be indispensable in this endeavor. These include the National Traffic System (NTS), *QST* and other publications to get the word out, and ability to raise needed funds. Finally, the League's good offices could be very valuable in resolving problems that arise.

We'd like to hear your views.

ASCII Modems... Computer Modems Packet Radio Modems...

--- Bell 202 Compatible, 1200 Bits Per Second ---

Rixon-Sangamo T202C & T202D & T202E

Recently the FCC opened the amateur bands to ASCII code transmission at speeds which far surpass the traditional RTTY rate of 45 Baud. For example, data transmission on 2 meters can now run at 1200 Baud, opening up new possibilities for computer-to-computer file exchange, packet radio, computer radio networks, automatic traffic handling between regions, and all sorts of new services which were unavailable previously.

To take advantage of this new service a station needs a computer (or at least a terminal), a modem and a radio. The modem converts the computer's digital signals to tones suitable for use on voice channels. There are many different modems currently in industrial use, but the Bell 202 type modem is quickly becoming the defacto standard for radio service at 1200 bps.

A Bell 202 modem is a frequency shift keying device which has a mark tone at 1200 Hz. and a space tone at 2200 Hz. These units generally operate in half-duplex mode (only one direction at a time) which is great for radio work, but a little inconvenient for telephone use. Some 202 units have a low speed reverse channel which allows full duplex signalling on a phone circuit. Since industry is currently adopting full-duplex 1200 bps modems, many 202 type units are appearing on the surplus market and can be purchased at very reasonable prices. There is no easier way to get your computer on the air.

In order to promote packet radio concepts and the greater use of ASCII on the air, Hank Magnuski, KA6M, purchased a large quantity of Rixon/Sangamo T202 modems which were being dumped by an industrial broker. Rixon/Sangamo is one of the leading and most reputable modem manufacturers in the country. These modems were built around 1970 from discrete components and are very rugged and serviceable. The manufactururer has been most cooperative in providing us with technical documentation on these modems, and we now have copies of the complete service manuals. The modems are being refurbished and checked out by Explorer Post 599, and various checks are made to insure that the units function. The prices are hard to beat:

T202D, 8 Card unit for 2-wire or 4-wire service, $75, T202D, 10 Card unit includes slow speed reverse channel, $85, T202C, 11 Card unit with reverse channel and auto answer, $90, T202E, 3 Card unit with transmit only capabilities, $50.

For more information contact: Chris Schellenberg, WB6WBK, (415) 324-4591, John Buonocore, KA6CUG, (415) 366-1658, Hank Magnuski, KA6M, (415) 854-1927.

Reprinted from the November/December edition of the *Homebrew Computer Club Newsletter*.

A NEW FACSIMILE TEST CHART has been published by the Institute of Electrical and Electronics Engineers as IEEE Standard 167A-1980. It is nearly identical in appearance to the 1975 edition except for some changes in type faces and better control of density ranges. Copies of the IEEE Facsimile Test Chart are available from the IEEE Service Center, 445 Hoes Ln, Piscataway, NJ 08854 for the price of $5 plus $2 shipping/handling.

REDUCING APPLE II RFI is the subject of an article, RFI: the F.C.C. and Your Apple which appears in the Fall 1980 edition of *Apple Orchard*, published by the International Apple Core, P.O. Box 976, Daly City, CA 95017. A parts kit is available from Apple Computer under part number 652-0152. It consists of two 0.1 µF capacitors, two solder lugs, and six ferrite beads.

ATV MAGAZINE NOW HAS a new address. Effective immediately all communications should go to ATV Magazine, 7391 West Highway 46, Elletsville, IN 47429. Thanks *HR Report*.

A MODEM POCKET GUIDE has been announced by Hayes Microcomputer Products Inc. The book, which is to be available through local computer stores, gives basics of data communications and explains how to expand your personal computer capabilities by adding a Hayes modem.

IF YOUR NEWSLETTER COPY has been stamped, then it's time to renew your membership.

THE DEAF AND THE TTY

Barry Strassler
Executive Director
Telecommunications for the Deaf, Inc.
814 Thayer Avenue
Silver Spring, Maryland 20910
TTY 301-589-3006

DEAF COMPUTER COMMUNICATIONS ENTER THE PICTURE

The costly long distance telephone bills hit the deaf people hard. Sociologically, their roots are not wholly confined to the neighborhood or to the immediate locale. It is not uncommon for their friendships and family ties to exist from coast to coast. It is theoretically possible for deaf families to embark on a cross-country motor trip and not spend a cent on motel fees, instead, staying at homes of people they know.

What this means is that they, just like everyone else, like to use the telephone to call up their friends and families. Long distance rates can be very costly, expecially with the benchmark of 150 words per minute voice rate versus the average typing speed of some 25 words per minute. For slow typists, the rate is even less. Thus, the electronic mail network comes very prominently into the picture.

In 1978, the Deaf Community Center (DCC) of Framingham, MA, aided by a government grant, pioneered the electronic mail network for the deaf. This set-up served approximately 50 deaf households in the Boston area. This was called the HERMES, and responsible for its implementation was the firm of Bolt, Beranek and Newman. The terminals being used were TI Silentype 700. The HERMES project losts its Federal funding recently, so the host DCC group switched to the GTE Telemail network. It is too early at this point to assess the Telemail progress.

Then in 1979, the SRI International landed a contract (with the TDI as the subcontractor) to develop the DEAFNET project in the Washington, DC area. The San Francisco Bay area was also used to a smaller extent. For a time, a three-city hookup was effected. The DEAFNET is similar to the HERMES but with one very important exception. The DEAFNET is compatible with the ASCII and Baudot terminals while HERMES does not accommodate the latter terminals. This DEAFNET contract is now in its third and final year of implementation. Currently underway is the establishment of a consortium to prolong the life of DEAFNET over and beyond the December 31, 1981 expiration date.

The Washington DEAFNET has succeeded while the San Francisco DEAFNET has not. Why? Zone toll call rates exist in the Bay area while the Washingtonians are blessed with a large metro toll area serving some 50 miles wide. No one in California wanted to incur long distance costs just to log into DEAFNET only to find no new messages waiting for them!

Such other systems are in use that can be likened to offshoots of the above mentioned networks. These are Virginia TTY and HEX.

The TDI is committed towards computer communications and is working with AMRAD to develop ideas toward this end.

THE 1981 INTERNATIONAL TELEPHONE DIRECTORY OF THE DEAF is now available from TDI. The new "blue book" lists residential, commercial, governmental and organizational TTY phone numbers throughout the United States and elsewhere. Membership dues are $10 for the first year and $5 for renewals. Write TDI, 814 Thayer Ave, Silver Spring, MD 20910 for application blank. Radio and computer amateurs with deaf TTY capability should list their numbers in the directory.

THE FCC HAS ALLOCATED CHANNELS FOR PAGING devices for the deaf, blind and other handicapped persons under PR Docket 79-315. The full text of the Commission's decision has not yet been released (at this writing) but the essence is that the FCC has designated 35.02 and 43.64 MHz for this new service. These frequencies would permit the operation of tactile paging units for the hearing impaired or conventional digital paging for others. Thanks K3JW.

AMSAT

Radio Amateur Satellite Corporation

P.O. BOX 27, WASHINGTON, D.C. 20044

Here are some excerpts from a letter written by AMSAT president Tom Clark, W3IWI to AMSAT life members:

I'm writing to you as a key supporter and Life Member of AMSAT to review the happenings of 1980, to describe our current activities and needs, to transmit to you my warmest Holiday Greetings, and to give you my New Year's prophesy on the directions of the future.

On New Year's day, 1980 we were involved in an intensive period of construction, integration and testing that, two months later, would culminate in the completion of Phase-IIIA. By late February, AMSAT's new spacecraft was ready for shipment to the European Space Agency (ESA) facility in Toulouse, France for launch vehicle integration and acceptance testing. All continued to go well, the hardware was shipped to Marburg for a memory transplant and final tuning of the antennas, and thence shipped to French Guiana for the launch. And then the unthinkable. BLACK FRIDAY. May 23, 1980 Ariane launch vehicle fails. Phase-IIIA is lost.

The gloom and despair and depression lasted for about a day when an interesting metamorphosis occurred. The "hard-core" of the technical workers, including both Jan King and Karl Meinzer began saying, "Damn it. We worked too hard to quit now. Let's try again." Like Phoenix, we have emerged from the ashes with an organization that is stronger than ever.

As we enter 1981, we have secured a ride for Phase-IIIB. The launch date is scheduled for February 24, 1982 on Ariane flight L-7. Hardware is already under construction in Marburg, Budapest and Washington. We are engaged in negotiations for a Phase-IIIC launch, with a date circa 1983. Thus two sets of Phase-III hardware are being constructed in parallel.

The people situation - Do you share the enthusiasm? Do you have some untapped talent that you can offer?

The money situation - The most serious problem is financial. The accelerated Phase-IIIB construction schedule, the decision to build Phase-IIIC in parallel, the requirements of the other activities, and the high inflation rate have upped the ante for the first half of 1981. Assuming that these new campaigns pick up momentum by next summer, we will need $70-100,000 to bridge the gap. If each of the Life Members gave just $50, we could make it through this critical period, especially if your contributions are applied against the ARRL Foundation and Hoover matching grants. Of course, some won't heed my plea, so I hope that you will be able to give more -- remember that the total needs for the next two years are $250,000.

It is you who help forge the future of AMSAT. We await your response, your vote of confidence, so that we may continue to expand amateur radio's horizons in space.

73,

Tom Clark - W3IWI

P.S. Thanks to you and AMRAD for your help and support. Tom

THE GEORGE WASHINGTON UNIVERSITY Continuing Engineering Education Program announces the following short courses to be presented in Washington, DC:

738 Voice Input-Output: Communications with Machines, Mar 23-26
252 Electromagnetic Interference and Control, Mar 23-27
597 Digital Telephony, Apr 6-10
551 Frequency Management, Apr 20-24
465 Data Communications Systems and Networks, Apr 27-May 1
760 Telecommunications: Voice and Data, Apr 27-29

For further information contact the Continuing Engineering Education Program, GWU, Washington, DC 20052, 202-676-6106 or 800-424-9773.

THE MICROPERIPHERAL CORPORATION has just introduced a new line of direct-connect modems designed to interface most popular computers and terminals to the telephone network. All units feature Bell 103 compatible operation in the originate or answer mode

The TRS-80Connection™ interfaces directly to the Model I and PMC-80 data/address bus. It can be connected directly to the keyboard to eliminate the need for the expansion interface or serial I/O RS-232 board.

The RS-232Connection™ plugs into the standard DB25 connector.

The ATARIConnection™ interfaces with the data I/O cable and connects in series with the 400 or 800 and cassette or disk drive.

The MICROCONNECTION™ for the APPLE™ employs a plug-in card to accomplish bus decoding and supplies an RS-232 compatible signal to the external MICROCONNECTION™.

Another option allows the MICROCONNECTION to be used with European systems (the EUROPEAN Connection). Tone frequencies are set to CCITT standards.

Prices start at $199.95. An AUTODIAL/AUTOANSWER option is available at $79.95, while the European option adds $29.95. The unit measures 7.75 in W x 5.75 D x 1.5 H and weighs less than a pound.

For additional information contact The MicroPeripheral Corporation, 4643 151st Place NE, Redmond, WA 98052, 206-881-7544.

TRS-80 is a trademark of Tandy Corp.
ATARI is a trademark of Atari, Inc.
APPLE is a trademark of Apple Computer.

TAB BOOKS, Blue Ridge Summit, PA 17214, 717-794-2191, announces publication of Radio Propagation Handbook (TAB Book No. 1146) by Peter N. Seveski, $17.95 hardbound-ISBN 0-8306-9949-X, $10.95 paperback-ISBN 0-8306-1146-0. Contents: Ground Wave Propagation, Ionospheric Waveguide Mode Propagation, High Frequency Ionospheric Propagation, Ionospheric Scatter Propagation, Microwave and VHF/UHF Propagation, Diffraction Propagation, Tropospheric Forward Scatter Propagation, Milimeter Wave Propagation. 504 pps.

RCA SOLID STATE, Box 3200, Somerville, NJ 08876 has announced price reductions on its CDP18S600 series of CMOS Microboard milliwatt computer systems as much as 47%. Write for details or call 201-685-6423.

HOWARD W. SAMS & CO, INC., 4300 West 62nd St, Indianapolis, IN 46268, 317-298-5400, has published 6502 Software Design by Leo J. Scanlon. In this book, the author gives you a working knowledge of the 6502 and the AIM computer. Scanlon is Documentation Manager for the Microelectronic Devices segment of Rockwell International. 272 pages. Softbound $10.50, ISBN 0-672-21656-6. Sams also has Design of VMOS Circuits, with Experiments by Robert T. Stone & Howard Berlin. 176 pps. Softbound $8.95, 0-672-21686-8.

AMRAD **Amateur Radio Research and Development Corporation**

Membership Application/Renewal

Mail to: Gerald Adkins, Treasurer
1206 Livingston St N
Arlington, VA 22205

See reverse for overseas mailing rates.

Dues:	Annual	Life
Regular	$12	$120
2nd in family	6	60
Full-time student	3	-

Please make checks payable to AMRAD.

Name____________________
Ham Call__________ Home Phone(   )__________
Address____________________
City, State__________ ZIP__________
I agree to support the purposes of the Corporation.

Class License__________ ☐ARRL Member
Have: ☐2-meter FM ☐RTTY
☐Computer model__________
Microprocessor type__________

*Signature*____________________

THE AMATEUR RADIO RESEARCH AND DEVELOPMENT CORPORATION is a technically oriented club of about 300 radio and computer amateurs. It is incorporated in the Commonwealth of Virginia and is recognized by the Internal Revenue Service as a tax-exempt scientific and educational organization.

THE PURPOSES OF THE CLUB are to: develop skills and knowledge in radio and electronic technology; advocate design of experimental equipment and techniques; promote basic and applied research; organize forums and technical symposiums; collect and disseminate technical information; and, provide experimental repeaters.

MEETINGS ARE ON 1st MONDAY of each month at 7:30 p.m. at the Patrick Henry Branch Library, 101 Maple Ave E, Vienna, VA. If the 1st Monday is a holiday, an alternate date will be announced in the *AMRAD Newsletter*. Except for the annual meeting in December, meetings are normally reserved for technical talks - not business.

THE WD4IWG/R REPEATER is an open repeater for data communications (including RTTY), voice and experimental modes. It is located at Tyson's Corner, McLean, VA and has excellent coverage. It features a semi-private autopatch available to licensed members. Frequencies are: 147.81 MHz input, 147.21 MHz output. The head of the technical committee is Jeff Brennan, WB4WLW, 7817 Bristow Dr, Annandale, VA 22003, phone 703-354-8541.

THE AMRAD NEWSLETTER is mailed monthly to all members and to other clubs on an exchange basis. Technical articles, new product announcements, news items, calls for papers and other copy related to amateur radio and computing are welcome. Honorariums at a rate of $10 per printed page ($20 maximum per author per issue) are paid for original material accepted. Classified ads are free to members. Commercial ad inquiries are invited. The editor reserves the right to reject or edit any portions of the copy. Items should be mailed by the 8th of the preceeding month to Paul L. Rinaldo, W4RI, Editor, 1524 Springvale Ave, McLean, VA 22101; phone 703-356-8918. Full permission for reprinting or quoting items appearing in the *AMRAD Newsletter* is granted provided that credit is given. Mailing is by 3rd Class bulk mail to U.S. addresses and 1st Class to Canada and Mexico. Overseas readers add 96¢ for surface or $7.80 for air mail to annual dues.

THE AMRAD MESSAGE SYSTEM is an S-100 Computerized Bulletin Board System on 703-734-1387, system operator Terry Fox, WB4JFI. Terry's home phone number is 703-356-8334. The system accepts 110, 300, 450 and 600 baud ASCII callers using Bell 103-compatible modems.

THE HANDICAPPED EDUCATION EXCHANGE (HEX) is operated by AMRAD for those involved in education and communications for the handicapped. It accepts both 110/300-baud ASCII and deaf TTY callers. on 301-593-7033. The sysop Dick Barth, W3HWN's home phone is 301-681-7372.

AMRAD OFFICERS for 1981 are:

Gerald Adkins, N4GA	Treasurer
Jeffrey Brennan, WB4WLW	Director Repeater Trustee
Tedd Riggs, KA4FYU	Librarian
Robert E. Bruninga, WB4APR	Director VP-Asia
Kenneth Coghill, WB4ZOH	1st Alt.
Terry Fox, WB4JFI	Director Vice Pres. Computer Trustee
William Pala, Jr., WB4NFB	Director Secretary
Paul L. Rinaldo, W4RI	Director President
Elton A. Sanders, Jr., WB5MMB	2nd Alt.

THE AMRAD LIBRARY is operated by Tedd Riggs, KA4FYU, 8402 Berea Ct, Vienna, VA 22180, phone 703-573-5067. Donations of technical books, magazines, manuals and catalogs are tax-deductible.

AMRAD IS AFFILIATED with the American Radio Relay League (ARRL), the Foundation for Amateur Radio, the Northern Virginia Radio Council (NOVARC) and The Mid Atlantic Repeater Council (T-MARC).

SPECIAL INTEREST GROUPS are formed from time to time. Currently we have SIG's on Deaf Communications and Spread Spectrum Communications. If you are interested in joining or forming a SIG, please contact Bill Pala, WB4NFB, 5829 Parakeet Dr, Burke, VA 22015; phone 703-323-8345.

TRAINING CLASSES are run as needed. Please discuss your training requirements with any Director.

AMATEUR RADIO RESEARCH AND
DEVELOPMENT CORPORATION
1524 SPRINGVALE AVENUE
MCLEAN, VA USA 22101

Nonprofit Organization
U.S. Postage
PAID
McLean, Virginia 22101
Permit No. 1511

Principles and Lessons in Packet Communications

LEONARD KLEINROCK, FELLOW, IEEE

Invited Paper

Abstract—After nearly a decade of experience, we reflect on the principles and lessons which have emerged in the field of packet communications. We begin by identifying the need for efficient resource sharing and review the original and recurring difficulties we had in achieving this goal in packet networks. We then discuss various lessons learned in the areas of: deadlocks; degradations; distributed control; broadcast channels; and hierarchical design. The principles which we discuss have to do with: the efficiency of large systems; the switching computer; network constraints; distributed control; flow control; stale protocols; and designers not yet experienced in packet communications. Throughout the paper, we identify various open issues which remain to be solved in packet communications.

I. Introduction

WHAT IS IT WE now know about packet communications that we did not know a decade ago? What made the problem difficult, and why were the solutions not immediately apparent to us in the late 1960's? Whereas the answers to these questions may entice the system designer (indeed, I, for one, delight in such investigations), why should the network user care a whit? To most users (and, alas, to many designers), communications is simply a nuisance and they would just as soon ignore those problems and get on with the "real" issues of information processing!

In this paper (and in conjunction with the other papers in this Special Issue of the PROCEEDINGS), we hope to answer some of these questions. We will identify the need for resource sharing, explain why the problem of efficient resource sharing is hard, and why it must be understood, review some of the lessons we learned (mostly from the three ARPA packet networks), and then, finally, state some principles which have evolved from the study and extensive use of packet communications.

II. Resource Sharing

A privately owned automobile is usually a waste of money! Perhaps 90 percent of the time it is idly parked and not in use. However, its "convenience" is so seductive that few can resist the temptation to own one. When the price of such a poorly utilized device is astronomically high, we do refuse the temptation (how many of us own private jet aircraft?). On the other hand, when the cost is extremely low, we are obliged to own such resources (we all own idle pencils).

An information processing system consists of many poorly utilized resources. (A resource is simply a device which can perform work for us at a finite rate.) For example, in an information processing system, there is the CPU, the main memory, the disk, the data channels, the terminals, the printer, etc.

Manuscript received March 24, 1978; revised June 16, 1978. This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-77-C-0272.

The author is with the Computer Science Department, University of California, Los Angeles, CA 90024.

One of the major system advances of the early 1960's was the development of multiaccess time-sharing systems in which computer system resources were made available to a large population of users, each of whom had relatively small demands (i.e., the ratio of their peak demands to their average demands was very high) but who collectively presented a total demand profile which was relatively smooth and of medium to high utilization. This was an example of the advantages to be gained through the smoothing effect of a large population (i.e., the "law of large numbers") [1]. The need for resource sharing is present in many many systems (e.g., the shared use of public jet aircraft among a large population of users).

In computer communication systems we have a great need for sharing expensive resources among a collection of high peak-to-average (i.e., "bursty") users [1]. In Fig. 1 we display the structure of a computer network in which we can identify three kinds of resources:

1) the terminals directly available to the user and the communications resources required to connect those terminals to their HOST computers or directly into the network (via TIPS in the ARPANET, for example—this is an expensive portion of the system and it is generally difficult to employ extensive resource sharing here due to the relative sparseness of the data sources;

2) the HOST machines themselves which provide the information processing services—here multiaccess time sharing provides the mechanism for efficient resource sharing;

3) the communications subnetwork, consisting of communication trunks and software switches, whose function it is to provide the data communication service for the exchange of data and control among the other devices.

The HOST machines in 2) above contain hardware and software resources (in the form of application programs and data files) whose sharing comes under the topic of time sharing; we dwell no further on these resources. Rather, we shall focus attention on those portions of the computer communications system where packet communications has had an important impact. Perhaps the most visible component is that of the communications subnetwork listed in item 3) above. Here packet communications first demonstrated its enormous efficiencies in the form of the ARPANET in the early 1970's (the decade of computer networks). The communication resources to be shared in this case are *storage capacity* at the nodal switches (these switches are called IMP's in the ARPANET), *processing capacity* in the nodal switches, and *communications capacity* of the trunks connecting these switches. Packet switching in this environment has proven to be a major technological breakthrough in providing cost effective data communications among information processing systems. Deep in the backbone of such packet-switched networks there is a need for long-haul high-capacity inexpensive communications, and it is here where we

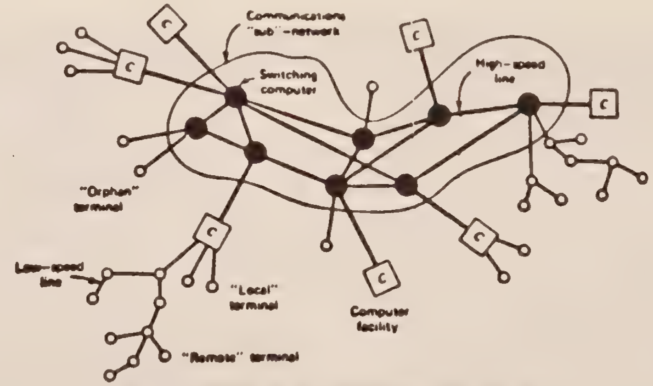

Fig. 1. The structure of a computer-communication network.

see the second application of packet communications for resource sharing in the form of satellite packet switching; elsewhere in this PROCEEDINGS [2] you will find a description of the SATNET, an ARPA-sponsored research network connected to the ARPANET. The third application may be found in the local access problem stated in item 1) above which also lends itself to the use of packet switching to provide efficient communications resource sharing; this takes the form of the use of a multiaccess broadcast channel in a local environment, commonly known as ground radio packet switching. Here too, ARPA has sponsored an experimental system, and its description may be found in this PROCEEDINGS [3]. The common element running through all these systems is the application of the smoothing effect of a large population to provide efficient resource sharing, an exquisite example of which is provided by packet communications.

Having described the environment and the resources of interest, let us now discuss the performance measures which permit us to evaluate the effort of resource sharing in a quantitative way. Indeed, there are basically four measures that both the system designer and network user apply in evaluating the service provided in a communications environment. These are *throughput*, *response time*, *reliability*, and *cost*. Before packet networks came into existence, the obvious solution for providing communications between two devices was to lease or dial a line between the two. In such a case the user was able to associate precise quantitative values to the four measures listed above. On the other hand when one attaches to a packet network, the user cannot get deterministic answers to the same quantities as he has in the past. He must accept probabilistic statements regarding throughput, delay, and reliability (and alas, sometimes even cost). Moreover the quantities so prescribed can seldom be measured in a straightforward fashion. This is the state of affairs to which we have evolved today! It is to the credit of those who developed packet communications in the last decade that the system design was carefully studied and well-analyzed *prior* to and during the systems implemention; this certainly has not, in general, been the history in the information processing industry.

III. Why the Problem Is Hard

Back in 1967, when the concept of the ARPANET was first taking form, we found ourselves entering the uncharted terrain of packet switching. Let us trace our initial confusion regarding that project briefly. Certainly, there existed at that time some communication networks, but they were mostly highly specialized networks with restricted goals. In the early 1960's Paul Baran had described some of the properties of data networks in a series of Rand Corporation papers [4]. He focused on the routing procedures and on the survivability of distributed communication systems in a hostile environment, but did not concentrate on the need for resource sharing in its form as we now understand it; indeed, the concept of a software switch was not present in his work. In 1968 Donald Davies at the National Physical Laboratories in England was beginning to write about packet-switched networks [5]; at around the same time, Larry Roberts at ARPA pursued the use of packet switching in an experimental nationwide network [6]. For a more complete history of the evolution of packet communications, see [7].

In the initial conception of a packet network, we identified some problems and looked to the technical literature for solutions to these problems. For example, how should one design the topology of a network, and how should one select the bandwidth for the various channels in such a network, and in what fashion should one route the data through the network, and what rules of procedure should two communicating processes adopt, and how much storage did one need at the multiplexing nodes of the network? These and many other questions confronted us. Indeed the general problem was how to achieve efficient resource sharing among a set of incompatible devices in a geographically distributed environment where access to these devices arose from asynchronous processes in a highly bursty fashion. Moreover, not only was the demand process bursty, it was also highly unpredictable in the sense that the instants when the demands arose and the duration of the demands were unknown ahead of time. Fortunately we were unaware of the enormity of the problems facing us and so we plunged ahead enthusiastically and with naive optimism. The remainder of this section describes why the problem was difficult, and in following sections we describe the lessons we learned and the principles we established in the development of packet communications. Our efforts have been well rewarded and the technology of packet communications has come of age and has proven itself to be a cost-effective technology.

We quickly found that many of our old techniques could not be directly applied to packet network design and that new techniques had to be developed; these new techniques turned out to be of great generality and have led to principles and to understanding which are sure to benefit us for many years to come. One of our first tasks was to develop tools which would allow us to analyze the performance of a given network. This involved evaluating the delay-throughput profile for networks. Basically, this is a queueing problem in a network environment. It had earlier been recognized [8] that the probabilistic complexities one encounters in computer networks are extremely difficult. One of the simplest analytical questions involving the solution of two nodes in tandem was first posed at that time in 1964 and has only been satisfactorily answered (in the queueing theoretic sense) within the past year [9]; this, note, is for the simplest problem. Indeed we have come to realize that an exact solution for the delay-throughput profile is probably hopeless in a complex network environment. Fortunately suitable approximations [1], [8] have been developed which permit one to predict the performance of given networks with a high degree of accuracy. More than that, these approximate tools allow us to expose and understand the phenomenological behavior of networks.

The astute reader will observe that the resource sharing problem stated above sounds very much like the problem faced in the design of time sharing systems. Surely, with time sharing, we are faced with the problem of sharing resources among asynchronous processes which behave in a bursty fashion. The major difference between the two problems, however, is that our problem exists in a *geographically distributed* environment which requires expensive communication resources in the communications and coordination functions. The implications here are strong. For example, when communication is cheap, then wide-band communications can be obtained with extremely small delays; such is the case, for example, within the resources of a local operating system connected together by a data bus. In a regional or nationwide network subject to the relatively expensive cost of telecommunications, we find that typical bandwidths are many orders of magnitude less than that in a local time-sharing environment, and the delays are many orders of magnitude greater. Furthermore, the control of these processes in the time sharing environment can be very tightly coupled if desired or left loosely coupled if there is sufficient reason; in the network environment we typically find our processes are very loosely coupled due to the difficulty of tightening the control between them (indeed, the inherent delay due to the finite speed of light is a fundamental limitation on the tight coupling of remote processes). The overhead in the time sharing system is variable and may be very high with poor system design (for example, thrashing) but may be made small with clever design. In the network environment, for a variety of reasons, we find that the overhead due to packet headers, control information and resource allocation tends to be relatively high. These comparisons are summarized in Table I.

Not only do we have extremes in communications cost between these two systems, we also have a significant difference in the reliability of that communications. Indeed, in the local time shared system, the process-to-process communication is usually assumed to be reliable and therefore the acknowledgment procedure (if any exists) is simple and tends to be invoked only under exceptional circumstances. On the other hand, in the distributed computer network environment, communications reliability is not assumed, and, therefore, an elaborate

TABLE I
ASYNCHRONOUS PROCESS-TO-PROCESS COMMUNICATION AND CONTROL COST COMPARISON BETWEEN LOCAL PROCESSES IN A TIME SHARED SYSTEM AND DISTRIBUTED PROCESSES IN A NETWORK

	Multiaccess Time-Shared Systems	Geographically Distributed Computer Networks
Typical bandwidth	megabytes/sec	kilobits/sec
Typical communications delay	fractions of a microsecond	tens to hundreds of milliseconds
Process-process coupling	tight to loose	typically loose
Overhead due to system control	variable (typically low)	variable (typically high)

acknowledgment procedure (often including timeouts and other defaults) is usually invoked. We are here reminded of the "two-army" problem. This is the problem where two blue armies are each poised on opposite hills preparing to attack a single red army in the valley. The red army can conquer either of the blue armies separately but will fail to defeat if both blue armies attack simultaneously. The blue armies communicate with each other over an unreliable communications system. The problem is to coordinate the two blue armies so that they will attack simultaneously. Let us assume that Blue Army 1 (B1) sends a message to Blue Army 2 (B2) indicating that they should jointly attack at noon the next day. Clearly B1 must await a positive acknowledgment from B2 that the command was properly received; were B1 to attack without such an acknowledgment, then the possibility exists that B2 did not receive the message correctly, in which case B1 is subject to the certain annihilation of his army. If B2 properly receives the command and acknowledges it, then he must await an acknowledgment of the acknowledgment from B1, for if B1 did not receive his acknowledgment then we know B1 will not attack and B2's attack will then be doomed to failure. The argument continues in a circular fashion where acks of acks of acks . . . are continually transmitted with no action ever being taken; the two blue armies can *never* get perfectly synchronized with certainty using this unreliable communications system.

We see therefore that the new culprit in resource sharing in a distributed environment is the fact that the resources are distributed and cannot easily be assigned to the demands. Indeed, in previous resource allocation problems, which often come under the name of scheduling algorithms, we made a big assumption, namely, that the scheduling information could be passed around for free. That is, the competing demands could organize themselves into a cooperating queue at no price. Unfortunately, in the distributed environment, the cost of organizing demands into a cooperating queue may be very large, and in one fashion or another nature will make you pay a price [10].

The problem of resource sharing in a distributed enivronment manifests itself in the routing control and flow control problems. The problem of flow control is to regulate the rate at which data crosses the boundary of the communications subnetwork (both into and out of the network). The problem of routing control is to efficiently transport the data (which has been admitted by the flow control procedure) across the network to its destination. In 1967 we were aware of the sophistication needed in the routing procedure, but were relatively ignorant of the need for effective flow control (see below). These two problems are difficult because we are dealing with a

control procedure in a distributed environment subject to random delays in passing that control information around in order to control the random demands. The purpose of both procedures is to efficiently use the network resources (IMP storage, IMP processing capacity, and communications capacity). In achieving this goal one must attempt to control congestion, route data around busy or defective portions of the network, and in general must adaptively assign capacity to the data traffic flow in an efficient, dynamic way. These are hard control problems and represent a class which has not been adequately studied up to and including the present time. We have come to learn that distributed control is a sophisticated problem. Below we return to the issues involved in distributed resource allocation and sharing. For the moment let us introduce some of the other sources of complexity in packet communications.

In any distributed communications system design one is faced with a *topological design problem*. The problem basically is, given a set of constraints to meet, find that topological design structure which meets these constraints at least cost. The field of network flow theory addresses itself to such problems and the salient feature of this theory is that most of its problems cannot be solved! To exhaustively search over all possible topological designs for a given problem is certainly not a solution since the number of possible alternatives to consider can easily exceed the number of atoms in the universe even for relatively small problems. (For example, if at some stage you must consider all permutations of 20 objects, then a computer would take more than 75 000 years to process all 20! cases even if it could examine one million cases per second.) Rather, a solution consists of elegant search procedures which are computationally efficient and which find the optimal topology for the given problem. Very few problems in network flow theory yield to such efficient algorithms. Rather, one gets around the combinatorial complexity naturally inherent in these problems by accepting suboptimal solutions. (Beware! A suboptimal solution to a problem is simply the result of a procedure which examines a subset of possible solutions and picks the best of those examined–this suboptimal solution may or may not be close to the optimal.) The trick here is to find efficient heuristic search procedures which come close to the optimal rapidly. Over the past decade, efficient procedures have been developed in many cases and new procedures are constantly being investigated for the topological design problem.

Another source of difficulty in the resource sharing problem is in defining the appropriate *performance measure*. For example, what is the capacity of a network? It is well-known in network flow theory that one can easily calculate the capacity (i.e., the throughput) between any two pairs of points. What is not straightforward is to evaluate the total data-carrying capacity of a network where throughput is measured in terms of messages successfully received at their destination. The difficulty comes about because the capacity of the network strongly depends upon the traffic matrix one assumes for the data flowing through that network. For example, if the traffic matrix were such that traffic passes only between immediate neighbors in the topological structure and in an amount equal to the capacity of the line connecting those two, then the network capacity would approach a value equal to the sum of all channel capacities in the network. This is clearly an upper bound to the capacity for any other traffic matrix. Since in general we do not know the traffic matrix for a network to be designed for future use, how is one to evaluate that capacity?

Yet another source of difficulty in the network problem is that of *interfacing* terminals and HOSTS to networks as well as one network to another network. For example, there is the general issue of *virtual circuit service* as opposed to *datagram service*. A virtual circuit service presents to the user a communications environment in which all functions appear as if she were directly connected between the two points communicating (i.e., an orderly and controlled flow is maintained whereby data is guaranteed to be correct upon delivery to the destination, comes out of the network in the same order in which it came in, and a flow control may be applied to that transmission to maintain efficient use of network resources and of terminal-HOST resources). A datagram service ensures none of these things and simply sends blocks of data (packets) across the network, not guaranteeing correct delivery (or delivery at all), not guaranteeing orderly flow in terms of sequencing (packets may arrive at the destination out of order), and not enforcing any flow control procedure at the process-to-process level. Which of those two services is desirable has become an issue of international proportions discussed elsewhere in these proceedings [7], [11]. Futhermore, the interconnection of two networks presents an enormous and rich variety of difficult problems. For example, should one apply flow control at the boundary of each network in a tandem chain of networks or should one apply flow control from the source HOST to the destination HOST across many networks simultaneously? If we consider the interconnection of networks with different packet sizes, we have the general problem of fragmentation whereby long packets get fragmented into smaller packets when crossing network boundaries. A variety of other very important issues arise in internetting (see [27] for a more detailed discussion of internetting).

Many of the problems we have just presented come about due to the distributed nature of our message sources and system resources. The problems created by distributed sources are very clearly seen in the environment of geographically dispersed message sources which communicate with each other over a common broadcast channel; in this case, access to the capacity of the channel must be carefully coordinated.

Thus in answering the question, "Why is the problem hard?" we have found the following sources of difficulty:

1) the analytic problems from queueing theory and the probabilistic complexity resulting therefrom;
2) the topological design problems from network flow theory and the combinatorial complexity resulting therefrom;
3) the price for coordinating and sharing resources in a geographically distributed environment (the new culprit) leading to problems of resource allocation, routing control, flow control, and general process-to-process communication problems.

IV. Lessons Learned

After a decade of experience with packet communications it is fair that we ask what lessons have we learned and what have we come to know about the needs of the user and the questions he would like to have answered. So far as the user is concerned we shall see as we step through the lessons learned below that he cannot insulate himself completely from the underlying technology of packet communications. Indeed the service he sees is quite different from that which he has with leased lines as mentioned above. Moreover, certain decisions will either be thrust upon him or accepted by him due to the nature of the service offered; if he is unaware of the consequence of setting

parameters in that decision-making process then he may seriously degrade the performance of the network due to his ignorance. Let us now list some of the lessons we learned and return to the principles in the following section.

A. Deadlocks

In [1], [12], and [13] we described in detail some of the deadlocks and degradations of which we have become aware. In this section we simply enumerate and sketch the details of the deadlocks. Simply stated, a *deadlock* (also commonly referred to as a *lockup*) is the unpleasant situation in which two (or more) competing demands have each been assigned a subset of their necessary resources; neither can proceed until one of them collects some additional resources which currently are assigned to the other and neither demand is willing to release any resource currently assigned to him. Deadlocks are one of the most serious system malfunctions possible, and one must take great care to avoid them or find ways to recover from them. It is ironic that flow control procedures by their very nature present *constraints* on the flow of data (e.g., the requirement for proper sequencing), and if the situation ever arises whereby the constraint cannot be met, then, by definition, the flow will stop, and we will have a deadlock! This is the philosophical reason why flow control procedures have a natural tendency to introduce deadlocks. In this section we briefly discuss four ARPANET deadlocks which have come to be known as: *reassembly* lockup; *store-and-forward deadlock*; *Christmas lockup*; and *piggyback lockup*.

Reassembly lockup, the best known of the ARPANET deadlock conditions (and one which was known to exist in the very early days of the ARPANET implementation), was due to a logical flaw in the original flow-control procedure. In the ARPANET, a string of bits to be passed through the network is broken into "messages" which are at most approximately 8000 bits in length. These messages are themselves broken into packets which are at most approximately 1000 bits in length. A message requiring more than one packet (up to a maximum of eight) is termed a multipacket message and each of these packets traverses the network independently; upon receipt at the destination node, these packets are "reassembled" into their original order and the message itself is recomposed, at which time it is ready for delivery out of the network. (A more complete description of the ARPANET protocols may be found in [1], [13].) Reassembly lockup occurred when partially reassembled messages could not be completely reassembled since the network through which the remaining packets had to traverse was congested, thus preventing these packets from reaching the destination; that is, each of the destination's neighbors had given all of their relay (store-and-forward) buffers to additional packets (from messages other than those being reassembled) heading for that same destination and for which there were no unassigned reassembly buffers available. Thus the destination was surrounded by a barrier of blocked IMP's which themselves could provide no store-and-forward buffers for the needed outstanding packets, and which at the same time were prevented from releasing any of their store-and-forward buffers since the destination itself refused to accept these packets due to a lack of reassembly buffers at the destination. The deadlock was simply that the remaining packets could not reach the destination and complete the reassembly until some store-and-forward buffers became free, and the store-and-forward buffers could not be released until the remaining packets reached the destination.

Store-and-forward deadlock is another example of a lockup that can occur in a packet-switched network if no proper precautions are taken [1], [13]. The case of "direct" store-and-forward lockup is simply a "stand-off." Let us assume that all store-and-forward buffers in some IMP *A* are filled with packets headed toward some destination IMP *C* through a neighboring IMP *B* and that all store-and-forward buffers in IMP *B* are filled with packets headed toward some destination IMP *D* through IMP *A*. Since there is no store-and-forward buffer space available in either IMP *A* or *B*, no packet can be successfully transmitted between these two IMP's and a deadlock situation results. One way to prevent the deadlock is to prohibit these packets in IMP *A* from occupying all those store-and-forward buffers which are needed by the packets coming in from IMP *B* (and vice versa) by the introduction of "buffer classes" as in [14]. This is accomplished by partitioning the buffers in a switch into classes, say, B_0, B_1, $\cdots$, B_k, where k is the longest path length in the network. A packet arriving at a switch from outside the net has access only to class B_0 buffers. When a packet arrives at a switch after having made k hops so far, it has access to class B_0, $\cdots$, B_k buffers, etc. Thus, the closer a packet gets to its final destination, the more access it has, and therefore the speedier its passage through the network. It can be proven [14] that this "buffer class" allocation will prevent direct store-and-forward lockup. "Indirect" store-and-forward lockup can occur when all store-and-forward buffers in a loop of IMP's become filled with packets all of which travel in the same direction (clockwise or counter-clockwise) and none of which are within one hop of their destination. Both store-and-forward lockup conditions are far less likely if, as in the ARPANET, more than one path exists between all pairs of communicating IMP's.

In December 1973, the dormant *Christmas lockup* condition was brought to life. This lockup was exposed by collecting measurement messages at UCLA from all IMP's simultaneously. The Christmas lockup occurred when these measurement messages arrived at the UCLA IMP for which reassembly storage had been allocated but for which no reassembly blocks had been given. (A reassembly block is a piece of storage used in the actual process of reassembling packets back to messages.) These messages had no way to locate their allocated buffers since the pointer to an allocated buffer is part of the reassembly block; as a consequence, allocated buffers could never be used and could never be freed. The difficulty was caused by the system first allocating buffers before it was assured that a reassembly block was available. To avoid this kind of lockup, reassembly blocks are now allocated along with the reassembly buffers for each multipacket message in the ARPANET.

Piggyback lockup is a deadlock condition which was identified by examining the flow control code and has, as far as we know, never occurred. This lockup condition comes about due to an unfortunate combination of intuitively reasonable goals implemented in the flow-control procedure. One of these goals, which we have already mentioned, is to deliver messages to a destination in the same order that the source received them. The other innocent condition has to do with the reservation of reassembly storage space at the destination. In order to make this reservation procedure efficient, it is reasonable that only the first multipacket message of a long file transfer be required

to make the reservation. The ARPANET flow control procedure will then maintain that reservation for a given file transfer as long as successive multipacket messages from that file are promptly received in succession at the source IMP. We have now laid the groundwork for piggyback lockup. Assume that there is a maximum of eight reassembly buffers in each IMP; the choice of eight is for simplicity, but the argument works for any value. Let IMP *A* continually transmit eight-packet messages (from some long file) to some destination IMP *B* such that all eight reassembly buffers in IMP *B* are used up by this transmission of multipacket messages. If now, in the stream of eight-packet messages, IMP *A* sends a single-packet message (not part of the file transfer) to destination IMP *B* it will generally not be accepted since there is no reassembly buffer space available. The single packet message will therefore be treated as a request for buffer allocation (these requests are the mechanism by which reservations are made). This request will not be serviced before the RFNM (an end-to-end acknowledgment from the destination to source) for the previous multipacket message has been sent. When the RFNM is generated, however, all the free reassembly buffers will immediately be allocated to the next multipacket message in the file transfer for efficient transmission as mentioned above; this allocation is said to be "piggybacked" on the RFNM. In this case, the eight-packet message from IMP *A* that arrives later at IMP *B* (and which is stored in the eight buffers) cannot be delivered to its destination HOST because it is out of order. The single-packet message that should be delivered next, however, will never reach the destination IMP since there is no reassembly space available, and, therefore, its requested reservation can never be serviced. Deadlock! A minor modification removes the piggyback lockup.

These various deadlock conditions are usually quite easy to prevent once they are detected and understood. The trick, however, is to expurgate all deadlocks from the control mechanism ahead of time, either by careful programming (a difficult task) or by some automatic checking procedure (which may be as difficult as proving the correctness of programs). Those deadlocks found in the ARPANET have, to the best of our knowledge, been eliminated.

B. Degradations

A degradation is just that, namely, a reduction in the network's level of performance. (Deadlocks are, of course, an extreme form of degradation which is why we discussed them in the separate section above.) For our purposes, we shall measure performance in terms of delay and throughput. In this section we discuss four sources of ARPANET degradation, namely: *looping* in the routing procedure; *gaps* in transmission; *single-packet turbulence*; and *phasing*.

Looping comes about due to independent routing decisions made by separate nodes which cause traffic to return to a previously visited node (or, in a more general definition, causes traffic to make unnecessarily long excursions on the way to its destination). Of course any reasonable adaptive routing procedure will detect these loops (through the build-up of queues and delays perhaps) and will then break the loop and guide the traffic directly on to its destination. However, the occurrence of loops does cause occasional large delays in the traffic flow and in some applications this is quite unacceptable. It is ironic that a remedy which was introduced in the ARPANET to reduce the occurrence of loops, in fact made them worse in the sense that whereas they occurred less frequently, when they did occur, they persisted for a longer time. Some loop-free algorithms have recently been published [15], [16].

The next degradation we wish to discuss is the occurrence of *gaps* in the message flow. These gaps come about due to a limitation on the number of messages in transit which the network will allow. Assume that between any source and destination, the network will permit n messages in flight at a time. If n messages are in flight, then the next one may not proceed until an end-to-end acknowledgment is returned back at the source for any one of the n outstanding messages. We now observe that if the round-trip delay (i.e., the time required to send a message across the network in the forward direction and to return its acknowledgment in the reverse direction) is greater than the time it takes to feed the n messages into the network, then the source will be blocked awaiting ack's to release further messages. This clearly will introduce gaps in the message flow resulting in a reduced throughput which we might classify as a mild form of degradation.

We now come to the issue of *single packet turbulence* as observed in the ARPANET. We note that "regular" single-packet messages in the early ARPANET were not accepted by the destination if they arrived out of order. Rather, they were then treated as a request for the allocation of one reassembly buffer. Therefore if, in a stream of single-packet messages, packet p arrived out-of-order (say it arrived after packet $p + 3$), then packets $p + 1$, $p + 2$, and $p + 3$ would all be discarded at the destination, and only after packet p arrived would a single packet buffer be allocated to message $p + 1$. This allocation piggybacked on the end-to-end ack for packet p, and when it arrived at the source IMP, it then caused a retransmission of the discarded packet $p + 1$ (which had been stored in the source). Of course any packets arriving at the destination after packet $p + 3$, but before $p + 1$ arrived in order, would themselves be discarded. When packet $p + 1$ finally arrived for the second time at the destination IMP it was then in order and this caused an allocation of a single-packet buffer to packet $p + 2$, etc. The net result was that only one packet would be deliverable to the destination per round-trip time along this path, had no packets been received out-of-order, then we would have been pumping at a rate close to n packets per round trip time (if the maximum number in transit n could fit into the pipe). Observe that once a single packet arrived out-of-order in this stream, then the degradation from n to 1 packets per round-trip time would persist forever until either some supervisory action was taken or until the traffic stream ceased and began again from a fresh start in the future. We refer to this effect as "single-packet turbulence," and it was observed in the ARPANET as described in [17]. The need to handle a continuous stream of traffic (e.g., packetized speech) was recognized some time ago and resulted in the definition of "irregular" packets known as type 3 packets (as contrasted to "regular" type 0 packets); these packets are allowed to be delivered out of order, receive no end-to-end acknowledgment, and are generally handled in a much more relaxed fashion.

The last degradation we discuss is known as "*phasing*." In a typical packet network, more than one resource is often required before a message is allowed to flow across that network. For example, some required resources may be: a message number; storage space at the source; storage space at the destination, etc. Tokens move around the network passing out

these resources in some distributed fashion. Phasing is the phenomenon whereby enough free tokens are available in the network to permit message flow, but, the proper mix of tokens is not available simultaneously at the proper location in the net. The delay in gathering these tokens represents a degradation [1], [18].

Fortunately, the degradations here described have been remedied in the ARPANET and in later networks.

C. Lessons of Distributed Control

We have had "lessons" in two areas of distributed control. The first has to do with flow control, and it is simply the observation that flow control procedures are rather difficult to invent and extremely difficult to analyze. The deadlocks and degradations referred to the in previous subsections were principally due to flow control failures (and occasionally routing control failures). To data there is no satisfactory theory or procedure for designing efficient flow control procedures, much less evaluating their performance, proving they contain no deadlocks, and proving that they are correct. Attempts in this direction are currently under way.

An important lesson we have learned with flow control is that a packet communications system offers an opportunity for passing data between two devices of (very) different speeds. We can effectively connect a slow-speed teletype to an enormously high-speed memory channel over a packet network and apply flow control procedures which protect the two devices from each other as well as protecting the net from both. Specifically, we must not drown the teletype with a flood of high-speed input, nor must we "nickel-and-dime" a high performance HOST to death with incessant interrupts, nor must we use the network as a storage medium for megabytes of data. Flow control mechanisms provide the means to accomplish this; the trick is to do it well.

The second area of distributed control in packet communications has to do with the routing control. The ARPANET, and many of the networks which have since based their design on the packet-switching technology which emerged from the ARPANET experiment, employ an adaptive routing procedure with distributed control. In such a procedure, routes for the data traffic are not preassigned but rather are dynamically assigned when they are needed according to the current network status. Control packets (called routing update packets) which describe the state of the network to some degree are passed back and forth between neighboring IMP's in some fashion and current queue lengths and congestion measures are added to these updates by each IMP. The ARPANET employs a periodic update routing procedure whose rate depends upon channel utilization and line speed. The updates passing between IMP's have no priority in competing for the processing capacity of the CPU at the IMP's but do have priority in the queue discipline feeding the output modems between IMP's. An important lesson learned is that giving low priority to the processing of routing updates appears to be advantageous since the processing load on the CPU is rather large and prevents the further dispatching of arriving packets to output queues [19]. Another routing lesson we have learned is that frequent updates cause background congestion in a network which may be intolerably high even in the absence of other data traffic; the update procedure and update rate must be carefully chosen. A number of alternatives to periodic updates have been suggested [1] including such things as aperiodic updating (send updates only when status information has crossed certain thresholds and then send it immediately); and purely local information for routing decisions based on queue lengths within a given node and knowledge of the current topology. Furthermore, unless care is taken, there is a tendency for looping to occur in these distributed control algorithms; looping can be prevented with more sophisticated algorithms [15].

One of the lessons which is now beginning to emerge is that the most important advantage of distributed control adaptive routing is its ability to *automatically* sense configuration changes in the network; these configuration changes may be planned or accidental as for example the result of a line or IMP failure. This is important for two reasons: first because configuration changes do happen often enough so that the requirement for a centralized control evaluating new routing tables based on the current configuration would be an enormously complex task from an administrative point of view; second because it is specifically at times of configuration changes when drastic network action must be taken and only then is the adaptive routing procedure really called upon to do serious work (it is not yet clear to what extent the routing algorithm should adapt to statistical fluctuations in traffic).

Without diminishing the result of these lessons, it is fair to say that the most significant lesson learned regarding routing is that it works at all. Perhaps one of the greatest successes of the ARPANET experiment was to show that a distributed control adaptive routing algorithm would indeed converge on routes which were sufficiently good. The difficulty in proving this lies in the fact that we are dealing with a dynamic situation in a distributed control environment with delays in the feedback paths for control information flow. The empirical proof that things do work has had an important impact on network design; indeed, these distributed algorithms are currently operating successfully in a number of packet networks.

D. Lessons from Broadcast Channels

As mentioned earlier, packet communications has found important applications in the areas of satellite packet broadcasting and in ground radio packet switching. In both environments we have a situation in which a common broadcast channel is available to be shared by a multiplicity of users. Since these users demand access to the channel at unpredictable times, we must introduce some access scheme to coordinate their use of the channel in a way which prevents degradations and mutual interference. In many of the schemes described [10] we have found that "burst" communications provides efficiencies over that of "trickle" transmission. By this we mean that when a data source requires access to the channel, it should be given access to the full capacity of that broadcast channel and not be required to transmit at a slow speed using only a fraction of the available bandwidth (see Section V-A on principles regarding "bigger is better").

In examining the recent literature, we find that a number of access schemes have been invented, analyzed, and published; for a summary of many of these access schemes, see [10]. We observe that these access schemes fall into one of three categories, each with its own cost. The first of these involves random access contention schemes whereby little or no control is exerted on the users in accessing the channel, and this results in the occasional collision of more than one packet; a collision destroys the use of the channel for that transmission. Such schemes as pure ALOHA, slotted ALOHA, and (to a much lesser extent) Carrier Sense Multiple Access fall into this category. At the opposite extreme, we have the static reservation access methods which preassign capacity to users thereby

creating "dedicated" as opposed to multiaccess channels. Here the problem is that a bursty user will often not use his preassigned capacity in which case it is wasted. Such schemes as Time Division Multiple Access and Frequency Division Multiple Access fall in this category. Between these two extremes are the dynamic reservation systems which only assign capacity to a user when he has data to send. The loss here is due to the overhead of implementing the demand access. Such schemes as Polling (where one waits to be asked if he has data to send), active reservation schemes (where one asks for capacity when he needs it), and Mini-Slotted Alternating Priority (where a token is passed among numbered users in a prearranged sequence, giving each permission to transmit as he receives a token) all fall in this category. Each of these schemes pays its tribute to nature as shown in Table II.

TABLE II
THE COST OF DISTRIBUTED RESOURCES

Access Method	Collisions	Control Overhead	Idle Capacity
Random access contention	Yes	No	No
Dynamic reservation	No	Yes	No
Fixed allocation	No	No	Yes

Unfortunately, at this point in time we are unable to evaluate the minimum price (i.e., a degradation to throughput and/or delay) one must pay for a given distributed multiaccess broadcast environment.

We have found that contention schemes are fundamentally unstable in that they have a tendency to drift into a congested state where the throughput decreases significantly at the same time the delay increases. Fortunately, however, we have been able to design and implement amazingly effective control schemes which stabilize these contention schemes [20]. Another lesson we have learned is that certain tempting ways of mixing two access schemes (e.g., taking a fraction of the traffic and a fraction of the capacity assigned to one access scheme, and using that capacity to handle that traffic using a second access scheme) does not give an improvement in the overall throughput-delay performance [10]. Furthermore we have found that certain capture effects exist in some of the contention schemes (e.g., a group of terminals may temporarily hog the system capacity and thereby "lock out" other groups for extended periods of time) and one must be wary of such phenomena [20].

We have also found that in a ground radio broadcast environment, a few buffers in each packet radio unit appear to be sufficient to handle the storage requirements [21]; this comes about largely due to the fact that our transceivers are half-duplex (i.e., they can either transmit or receive, but not both, at a given time). We can show (see Section IV-E) that dedicated broadcast channels have an inherent advantage over dedicated wire networks in a large (many-user) bursty store-and-forward environment [22]. Moreover, we have investigated the optimal transmission range for ALOHA networks and have found that those broadcast networks can be made quite effective when the traffic is not bursty; indeed this optimal range is chosen so that the channel utilization in the resulting local ALOHA system is $1/2e$ and then those networks need only $\sqrt{e}$ more capacity than the corresponding M/M/1 network [22].

Lastly we point out that perhaps one of the first applications of broadcast radio access schemes will be to implement these access schemes on wire networks (for example, coaxial cables or fiber-optics channels) in a local environment; an example of such an implementation is the Ethernet [23].

E. Hierarchical Design

As N (the number of nodes in a network) grows, the cost of creating the topological design of such a network behaves like N^E where E is typically in the range from 3 to 6. Thus we see that topological design quickly becomes unmanageable. Secondly, we note that as N grows, the size of the routing table in each IMP in the network grows linearly with N and this too places an unacceptable burden on the storage requirements within an IMP. In addition, the transmission and processing costs for updating such large tables is prohibitive. Third, even were the design possible, the cost of the lines connecting this huge number of nodes together grows very quickly unless extreme care is taken in that design. In all three cases just mentioned, one finds that the use of hierarchical structures saves the day. In the design case, one may decompose the network into clusters of nodes, superclusters of clusters, etc., designing each level cluster separately. This significantly reduces the number of nodes involved in each subdesign, thereby reducing the overall design cost significantly. For example, a 100-node net would have a cost on the order of $100^4 = 10^8$ (for $E = 4$), whereas a 2-level hierarchical design with 5 clusters would cost on the order of $5(20)^4 + 5(4) \leq 10^6$, yielding an improvement of three orders of magnitude! The same approach may be used in routing, where names of distant clusters, rather than names of distant nodes, are used in each routing table, thereby reducing the table length down from N to a number as small as $e \ln N$ giving a significant reduction [24]. For example, a 1000-node net would give a 50-fold reduction in the routing table length when hierarchical routing is used.

In [22] we discuss the overall effect and gain to be had in the use of hierarchically designed wire networks and broadcast networks. For example, we can show that in a bursty dedicated broadcast environment, the use of hierarchical network structures (even with fixed allocation schemes) yields a system cost which is proportional to $[\log M]^2$, where M is the number of users. Comparing this to the case of wire networks where the cost is proportional to the $\sqrt{M}$, we see the significant advantages that broadcast channels have over wire networks in a bursty environment when hierarchical structures are allowed. We can see this intuitively since we assume that the cost of a broadcast channels is proportional only to capacity, but is independent of distance; if we properly select the transmission range, then the broadcast capacity can be reused spatially (i.e., it can be used independently and simultaneously in more than one area). Further, it can be shown that a 2-level hierarchy using random access in the lower level and dedicated channels in the upper level can be quite efficient in a broadcast environment; this is true since the lower level has gathered together enough traffic so that it is no longer bursty when delivered to the upper level (recall that dedicated channels do well with nonbursty traffic)

V. PRINCIPLES ESTABLISHED

This section is really a continuation of the last since there is a somewhat fuzzy boundary between lessons and principles. Indeed, one might accept the pragmatic definition that a principle is a lesson you had to learn twice.

A. Bigger is Better

The law of large numbers states that a large collection of demands presents a total demand which is far more predictable

than are the individual demands. We are thus led to the consideration of large shared resources (large in the sence that we increase both the number of users—or the load presented by each user—and the capacity of the resources). Furthermore it is easy to show that the performance improves significantly as we make our systems larger. In particular we can show that a small system whose capacity is C operations per second and whose throughput is J jobs per second (with each job requiring an average of K operations per job) performs A times as slowly (i.e., the response time is A times longer) as a system whose capacity is AC and whose throughput is AJ. The lesson here is very clear, namely that bigger systems perform far better than smaller ones [25]. This is a statement about performance and not one about cost. Indeed if one is talking about communication channel capacity, then one usually also gains through an economy of scale due to the tariff pricing structure as presented by the common carriers. All the more reason, therefore, to concentrate more and more traffic on ever larger channels to gain both cost and efficiency in performance; of course one must be careful not to abuse any "resale" restrictions. Moreover, our lesson about burst communications tells us that in sharing this large channel dynamically, one should provide the full capacity to a single user on demand, rather than to preallocate fractions of the capacity on a permanent basis (omitting consideration of such channel-sharing schemes as spread-spectrum).

The "bigger is better" principle may not apply to the case of stream traffic (defined as real-time traffic which requires a low delay and moderately large throughput requirement—an example being packetized speech). Indeed, an unresolved issue recently raised by Dr. Robert E. Kahn (Editor of this Special Issue) is how effective it would be to handle stream traffic by dividing each trunk into a multiplicity of medium-capacity channels which may then be linked together to form a stream traffic path. We are currently looking at this issue.

B. The Switch

Our second principle has to do with the use of a software switch at the nodes of a network. The principle here is that it pays to place intelligence at the switching nodes of a network since the cost of that intelligence is decreasing far more rapidly than the cost of the communications resource to which it is attached. The idea is to invest some cost in an intelligent switch so as to save yet greater cost in the expensive communications resource. The ability to introduce new programs, new functions, new topologies, new nodes, etc., are all enhanced by the programmable features of a clever communications processor/multiplexer at the software node.

C. Constraints

The principle here is simply, "constraints are necessary and often are evil." Indeed some of the constraints we have seen are sequencing, storage management, capacity allocation, speed matching, and other flow and routing control functions. These "natural" constraints render us vulnerable to dangerous deadlocks and degradations. As mentioned above, if the constraint cannot be met due to some possibly unfortunate accident, than the system will stop all flow. If one is slow in meeting the constraints, then that represents a delay-throughput degradation. As a result of this principle, we see that it behooves us to provide sufficient resources in the network which then allow us to be more relaxed about assigning them. That is, the more precious is a given resource, the tighter we are in allocating it to a demand, the more likely we are to run into a deadlock or degradation. With an ample resource, we can be more cavalier in assignment and even renege on the assignment if necessary, assuming that a backup facility (in the form of an ample resource elsewhere in the network) is provided.

D. Distributed Control

The principle here is that one must pay a price to nature for organizing a collection of distributed resources into a cooperating group. We have not yet established what that minimum price is, but we have classified the price in the form of collisions, control overhead, and idle capacity.

E. Flow Control

The "principle" here is that flow control is a critical function in packet communications and we are still naive in the invention and analysis of flow control procedures. Hopefully, cleaner code and cleaner concepts will simplify our ability to design and evaluate flow control procedures in the future. There is a "miniprinciple," which seems to be emerging from our preliminary studies [26] which states that if one wants to maximize the power in a network at fixed cost, where power is defined as throughput divided by response time, then under simple statistical assumptions on the flow, one should operate at a point where the throughput delivered is half the maximum possible and the response time is then twice the minimum (no-load) response time.

F. Stale Protocols

In our experiments in the ARPANET, the SATNET, and the packet radio network, we have occasionally attempted to adopt a protocol from one network directly over into a new network. We have found that this is a dangerous procedure and must be carefully analyzed and measured before one adopts such a procedure. Indeed, the use of old protocols in a new environment is dangerous. For example, we found that the use of the ARPANET-like RFNM end-to-end protocol was extremely wasteful of channel capacity and resulted in a capture effect between pairs of users when used in the SATNET. In a 2-user Time Division Multiple Access scheme (in which odd-numbered slots are permanently assigned to user A and even-numbered slots to user B), user A could prevent B from sending any data if he simply started transmitting first in each of his slots since this would require B to devote all of his slots to returning RFNM's to A. Time in the SATNET is divided into fixed length slots (of 30-ms duration). A slot is used for a single packet transmission even if the packet itself is tiny, as is the case of a RFNM. This inefficiency does not exist in the ARPANET since no extra bits are stuffed into ARPANET packets to artificially increase their size. Indeed, gateways have been introduced between the ARPANET and SATNET which renders these nets independent of each other's protocols and formats [2].

G. Inexperienced Designers

It is important that users recognize the difference in function, performance, and operation of a packet network as opposed to a leased line. Certain decisions regarding the parameter settings in any process-to-process communication are often left up to the user of a packet network; for example the buffer allocation he provides in his HOST to accept data from another process communicating through the network with his HOST is a decision often left to the system user. If his buffer allocation is too small, he may degrade the apparent performance of the network to an unacceptably low degree; this comes about, not because the network is slow, but rather because his allocation was too small. The principle here is that if one leaves design decisions in the hands of the users (or even network designers) then those individuals must be informed as to the effect of their decisions regarding these parameter settings; they cannot be expected to understand the consequence of their actions without being so informed.

VI. Conclusions

The purpose of this paper has been to boil down a decade of experience with packet communications and from this to extract some lessons and principles we have established. We have succeeded only in part in this endeavor; the field is still moving rapidly and we are learning new things each day. Indeed, in addition to lessons and principles, we have identified a number of open issues which require further study. Aside from the meager principles we stated in the preceding section, we feel it is necessary to make some concluding statements. First we feel that one must view packet communications as a system rather than as a trivial leased line substitute. The use of packet communications offers opportunities to the informed user on the one hand and sets traps for the naive user on the other. It is necessary that the overriding principles which we have established and others which we have yet to establish be well understood by the practitioners in the field. We must continue to learn from our experience, and alas, that experience is often gained through mistakes observed rather than through clever prediction. In all of our design procedures we must constantly be aware of the opportunity to share large resources among large populations of competing demands. We must further be prepared to incorporate new technologies and new applications as they arise; we cannot depend upon "principles" as these principles become invalid in the face of changing technologies and applications.

Lastly, we must point out that the true sharing of processing facilities in the network (i.e., the HOSTS) has not yet been realized in modern day networks. One would dearly love to submit a task to a network, ask that it be accomplished in the most efficient fashion, and expect the network to find the most suitable resources on which to perform that task. Currently, one must specify on which HOST his program should be stored, where his job should be executed, where to store his results, at which location his results should be printed, and specify when all this must happen. The next phase of networking must address this general question of automatic resource sharing among HOSTS in a distributed processing environment. Perhaps in the next special issue on packet communications we will be in a position to identify lessons and principles for true resource sharing of this type.

References

[1] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications.* New York: Wiley Interscience, 1976.

[2] I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose packet satellite networks," this issue, pp. 1448-1467.

[3] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," this issue, pp. 1468-1496.

[4] P. Baran, "On distributed communications," RAND Series Reports, Rand Corporation, Santa Monica, CA, Aug. 1964.

[5] D. Davies, "The principles of a data communication network for computers and remote peripherals," in *Proc. IFIP Congress '68*, Edinburgh, Scotland, pp. 709-714; Aug. 1968.

[6] L. G. Roberts, "Multiple computer network development to achieve resource sharing," in *Proc. ACM Symp. Operating Syst.*, Gatlinburg, TN, 1967.

[7] L. G. Roberts, "The evolution of packet switching," this issue, pp. 1307-1313.

[8] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay.* New York: McGraw-Hill, 1964, out of print. Reprinted by Dover Publications, 1972. (Published in Russian, 1970, Published in Japanese, 1975.)

[9] O. J. Boxma, "On a tandem queueing model with identical service times at both counters, I.," University Utrecht, Dept. of Mathematics, Preprint No. 78, Mar. 1978.

[10] L. Kleinrock, "Performance of distributed multi-access computer-communication systems," in *Proceedings of IFIP Congress '77* Toronto, Canada, pp. 547-552; Aug. 1977.

[11] L. Pouzin and H. Zimmerman, "A tutorial on protocols," this issue, pp. 1346-1370.

[12] L. Kleinrock, "ARPANET lessons," in *Proc. Int. Conf. Communications*, Philadelphia, PA, pp. 20-1–20.6; June 1976.

[13] R. Kahn and W. Crowther, "Flow control in a resource sharing computer network," in *Proc. 2nd IEEE Symp. Problems in Optimization of Data Communication Systems*, Palo Alto, CA, pp. 108-116, Oct. 1971, (also reprinted in *IEEE Trans. Communications*, pp. 539–546; June 1972).

[14] E. Raubold and J. Haenle, "A method of deadlock-free resource allocation and flow control in packet networks," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 483-487; Aug. 1976.

[15] W. Naylor, "A loop-free adaptive routing algorithm for packet switched networks," in *Proc. Fourth Data Communications Sym.*, Quebec City, Canada, pp. 7.9–7.14; Oct. 1975.

[16] A. Segall, P. M. Merlin, and R. G. Gallager, "A recoverable protocol for loop-free distributed routing," in *Pro. Int. Conf. Communications*, Toronto, Canada, vol. 1, pp. 3.5.1-3.5.5; June 1978.

[17] H. Opderbeck and L. Kleinrock, "The influence of control procedures on the performance of packet-switched networks," in *Nat. Telecommunications Conf. Record*, San Diego, CA., pp. 810-817; Dec. 1974.

[18] W. Price, "Simulation of packet-switching networks controlled on isarithmic principles," in *Proc. Third Data Communication Symp.*, St. Petersburg, FL, pp. 44–49; Nov. 1973.

[19] W. Naylor and L. Kleinrock, "On the effect of periodic routing updates in packet-switched networks," in *Nat. Telecommunications Conf. Record*, Dallas, TX, pp. 16.2-1–16.2-7; Nov. 1976.

[20] L. Kleinrock and M. Gerla, "On the measured performance of packet satellite access schemes," in *Proc. Fourth Int. Conf. Computer Communication*, Kyoto, Japan, Sept. 1978.

[21] F. Tobagi, "Performance analysis of packet radio communication systems," in *Nat. Telecommunications Conf. Record*, pp. 12.6-2-12.6-7; Dec. 1977.

[22] G. Akavia, "Hierarchical organization of distributed packet-switching communication systems," Ph.D. Dissertation, Computer Science Department, Univ. of California, Los Angeles, Mar. 1978.

[23] R. Metcalfe and D. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, no. 7; pp. 395–404, July 1976.

[24] L. Kleinrock and F. Kamoun, "Data communications through large packet-switching networks," in *Proc. Int. Teletraffic Congress*, Sydney, Australia, pp. 521-1–521-10; Nov. 1976.

[25] L. Kleinrock, "Resource allocation in computer systems and computer communication networks," in *Proc. of IFIP Congress '74*, Stockholm, Sweden, pp. 11–18; Aug. 1974.

[26] —, "On flow control," in *Proc. Int. Conf. Communications*, Toronto, Canada pp. 27.2-1 to 27.2-5, June 1978.

[27] V. G. T. Cerf and P. Kirstein, "Issues in packet network interconnection," this issue, pp. 1386-1408.

I-FRAME de VE3PKT

·--· -·- -

Newsletter of the

HAMILTON & AREA PACKET NETWORK

2391 Arnold Crescent, Burlington, Ontario,
Canada, L7P 4J2